Lane Detection using OpenCV: A Video-based Approach

Chirag Dhoka Jain
Final Project
course - ***EECE5330***
dhokajain.c@northeastern.edu
April 24th, 2024
Northeastern University

Keval Visaria
Final Project
course - ***EECE5330***
visaria.k@northeastern.edu
April 24th, 2024
Northeastern University

Tejasri Kasturi
Final Project
course - ***EECE5330***
kasturi.t@northeastern.edu
April 24th, 2024
Northeastern University

Veditha Gudapati
Final Project
course - ***EECE5330***
gudapati.ve@northeastern.edu
April 24th, 2024
Northeastern University

***Abstract*—In this project we leverage we present application to lane detection of the road in the streets based on the some image processing techniques, we will evaluate the performance of method in different type of the datasets such as straight and curve road in different lightning condition. we proposed the results of this application for most of the image datasets in the experiment and result section. Lane detection and tracking technique are commonly used for a vehicle to navigate autonomously on the road. The common lane detection process includes the preprocessing, feature extraction and lane detection and tracking. The pre-processing process includes the process of determining the region of interest (ROI) as stated that ROI important to lessen the detection of false lane and improve the computational efficiency.**

***Index Terms*—Image processing, Computer vision, Thresholding, Color spaces, Region of Interest(RoI), Convolutional Neural Networks(CNN), Robustness**

I. Introduction

In recent years, the evolution of computer vision technology has brought about transformative changes across various domains, particularly in the autonomous driving and road safety industry. Central to the functionality of autonomous driving systems is the task of lane detection, which has a crucial role in assisting vehicle navigation and mitigating risks of road accidents. Lane detection, a foundation of Advanced Driver Assistance Systems (ADAS) [5], includes a spectrum of functionalities ranging from preemptive alerts when a vehicle steers out of its lane to facilitating lane maintenance, overtaking maneuvers, and automated cruising.

The vital importance of lane detection algorithms stem from their role in safeguarding the well-being of passengers and the integrity of the vehicle itself. Nevertheless, the intricate nature of real-world driving scenarios makes lane recognition challenging. As highlighted in [3], the future trajectory of the automotive industry is dependent upon the comprehension of real-time driving scenarios, a vision that receives increasing support from both academic and industrial spheres.

While substantial developments have been made in enhancing road experiences, particularly concerning lane detection, there remains a pressing need for further refinement to overcome existing challenges and enhance driver convenience. Accurate detection of lane markings empowers autonomous vehicles to interpret their surroundings adeptly, thereby facilitating safe and efficient navigation. There are numerous lane identification techniques available, each with unique benefits and drawbacks based on the driving environment. Principal among the requirements for effective lane detection algorithms is their real-time responsiveness and robustness in adverse road conditions, where discerning lane markings becomes taxing. Notably, [4] explores the integration of diverse color spaces and lane shape features to address the complexities inherent in recognizing varied lane markings on roads.

The transformative potential of these advancements extends beyond mere technological innovation, promising profound societal impacts and indicating substantial enhancements in the efficiency, convenience, and safety of our road networks and transportation systems.

To harness the potential of these innovations, our proposed approach makes use of a range of techniques, including perspective transformation, edge detection, histogram analysis, and polynomial fitting (Bird's-Eye View). Through the judicious application of these techniques and analysis of video footage captured, our objective is to develop robust algorithms capable of lane detection under various environmental conditions, such as varying lighting conditions, road surface textures, and traffic densities.

II. Related Work

A comprehensive review of existing methodologies for lane detection reveals a diverse range of approaches, each with its unique advantages and challenges. This section examines various traditional and deep learning-based techniques, focusing on their application in real-time systems and under different environmental conditions. Highlighting innovations and limitations within each method provides a foundational context for the development and optimization of our project.
In[6] presents a real-time monocular vision-based lane departure warning system that effectively handles varying lighting conditions using Otsu's threshold method and Hough Transform. The key innovation lies in its ability to adjust to different light levels, ensuring reliable lane detection and providing vehicle position warnings. This work is particularly pertinent to projects aiming to improve lane detection robustness across diverse driving conditions .
Explores[7] various image processing techniques such as OpenCV, Gaussian Blur, Masking, Canny Edge Detection,

and Hough Transform for real-time lane detection on two-lane roads. The system was tested with a vision sensor installed in a vehicle, capturing video at different times of the day to assess performance variations. Key findings include optimal detection at certain times with daylight and identified improvement needs for night-time and shadow conditions. This approach aligns with projects aiming to enhance autonomous navigation without deep learning, focusing on real-time, low-cost, and efficient image processing methods.
Surveys[4] various techniques in lane detection, comparing traditional image processing methods with advanced deep learning models. It discusses the reliability and processing challenges of each method under different road conditions, evaluating their effectiveness in real-time applications. Deep learning approaches offer higher accuracy and adaptability to complex environments, whereas non-deep learning techniques remain valuable due to their efficiency, lower computational demands and cost-effectiveness.

Our project aims to evaluate various color spaces for lane detection and to conduct a comparative analysis between traditional methods and deep learning approaches, thereby assessing their respective efficiencies and accuracies in different scenarios.

III. Methods

The methodology section outlines a sequence of processes applied to video frames for lane detection, using OpenCV. Our system is designed to handle video and 2-D image input. Below, we expand on each step involved in detecting lanes from these frames.

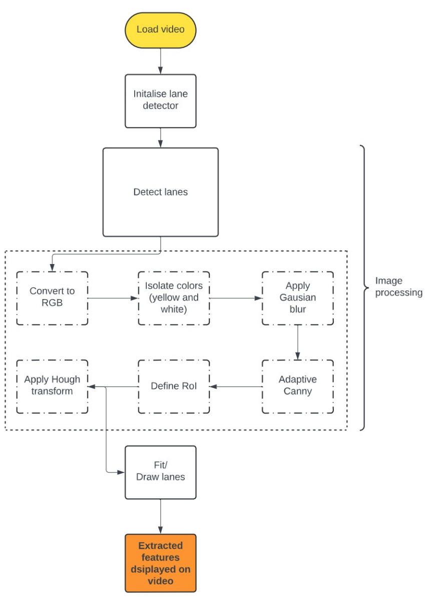

Fig. 1. Traditional algorithm

A. Color Filtering

The initial step in processing the frames involves converting the color space from BGR (Blue, Green, Red), which is the default format for images in OpenCV, to RGB (Red, Green, Blue). This conversion aligns with human visual perception, making subsequent color-based operations more intuitive. After conversion, the frame is processed to isolate yellow and white colors, typical of road lane markings. Separate binary masks are created for each color using specific color range thresholds. These masks are then combined using a bitwise OR operation, which merges all detected lane markings into a single binary image. This filtered output isolates relevant lane information from irrelevant background details.

B. Region of Interest (ROI) Extraction

To focus the processing on likely lane regions and to reduce computational waste, a triangular region of interest (ROI) is dynamically defined based on the frame's resolution. The vertices of the triangle are calculated to pinpoint the lower half of the frame, concentrating on the roadway ahead. This strategic reduction in the analysis area decreases noise and irrelevant details, significantly enhancing processing efficiency.

C. Edge Detection

Following color filtering, the resultant image is converted to grayscale to simplify the data, focusing solely on intensity values. This grayscale image is then subjected to Gaussian blurring, which helps in reducing high-frequency image noise that could disrupt edge detection. Subsequently, edges are detected using the Canny edge detector, an algorithm that identifies areas of high gradient in the image, indicative of boundaries. The adaptive nature of the applied Canny detector adjusts its thresholds based on the image content, ensuring robust detection across varying lighting conditions.

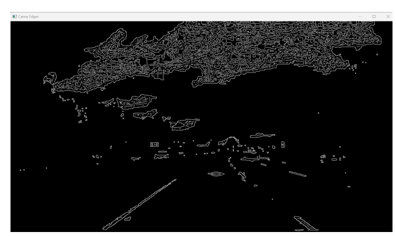

Fig. 2. Output image using canny edge detection

D. Hough Transform

The Hough Transform is a feature extraction technique used here to identify straight lines within the edge-detected image. This method translates each point in the image to a series of curves in a parameter space, and the intersections of these curves correspond to potential lines in the image space. The detected lines are then analyzed to determine their orientation and slope, categorizing them as either left or right lane markings based on predefined slope thresholds.

E. Line Fitting

For each set of classified line segments, a line fitting algorithm using least-squares regression is employed. This mathematical method optimizes the fit of a line to the set of points, reducing the sum of the squared differences between the line and the points. This process smooths and unifies disparate line segments into coherent lane lines, providing a continuous visual representation of each lane boundary.

F. Lane Averaging and Drawing

The $LaneDetector$ class plays a crucial role in both averaging and visually representing the detected lane lines. It employs a buffer to store recent detections of lane lines across multiple frames. By averaging these detections over a specified window, the system effectively smooths out anomalies and inconsistencies caused by transient discrepancies in individual frame analysis. This averaging not only stabilizes the visual output but also improves the accuracy and reliability of lane representations. Once averaged, the lane lines are graphically rendered onto the original frame using OpenCV's line drawing functions. This step is vital for providing a clear visual representation of the detected lanes. It allows the lines to be drawn with precision over the roadway in the video, offering a real-time overlay that clearly demonstrates the lane boundaries. This integrated display aids in the visual assessment of the lane detection system's performance and provides valuable feedback for further adjustments and improvements.

G. Display Output

The composite frame, which superimposes the detected lanes onto the original video frame, is created using a weighted sum approach. This approach maintains the background context while highlighting the lanes, and the result is displayed in real-time to provide an immediate visual reference of the lane detections.

H. Advanced Lane Detection System

The advanced lane detection system enhances the base methodology by integrating sophisticated computer vision techniques to improve the detection accuracy and robustness of lane markings on roadways. Below, we provide an in-depth explanation of each step involved in this system.

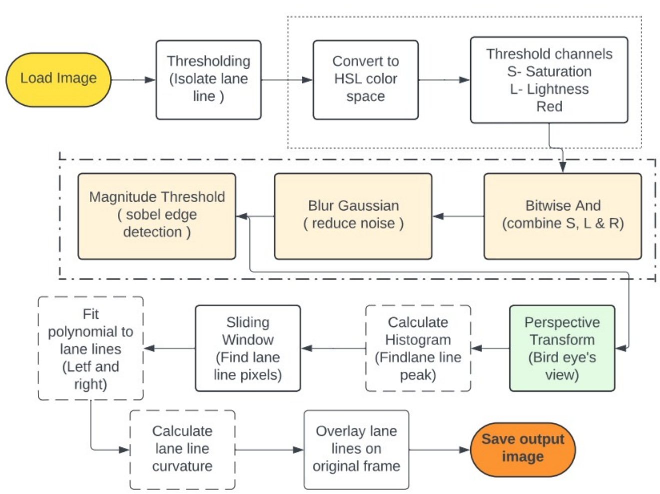

Fig. 3. Detailed workflow

1) Color Space Conversion:
The image is converted from the BGR (Blue-Green-Red) color space to HLS (Hue-Lightness-Saturation). This conversion is crucial as HLS separates image luminance from color information, which is more effective for handling variations in lighting and color saturation when detecting lane markings.

2) Edge Detection:
The Sobel operator is applied specifically to the Lightness channel of the HLS image. This operator detects edges by calculating the gradient of image intensity at each pixel, highlighting areas with sharp changes that are likely to be edges of lane lines. This technique is sensitive to both horizontal and vertical changes, making it ideal for capturing the varied orientations of lane lines.

3) Binary Thresholding on Saturation:
Thresholding on the Saturation channel isolates regions with pure colors, which are indicative of lane markings typically painted in solid white or yellow. This step filters out less saturated and dull colors that do not contribute to the lane markings.

4) Binary Thresholding on Red Channel:
The Red channel is also thresholded to identify high values that correspond to the colors of lane lines. Since lane markings often contain white, which has a high red component in RGB space, this channel is effective in further distinguishing them from other elements in the scene.

5) Combining Binary Images:
The binary images from the Saturation and Red channel thresholding are combined using a bitwise AND operation. This combination helps in reducing noise by ensuring that only pixels which are identified as lane markings in both binary images are retained, enhancing the clarity of lane line detection.

6) Incorporating Edge Detection:
The edge-detected image is merged with the combined binary image using a bitwise OR operation. This integration ensures that the final binary image includes comprehensive features of lane lines, both their solid color regions and distinct edges, facilitating more accurate detection.

7) Histogram and Lane Location:
A histogram of the binary image is computed to analyze the distribution of white pixels across the image width. Peaks in this histogram represent likely lane line positions, serving as a starting point for more precise localization using the sliding window technique.

8) Sliding Window Technique and Curve Fitting:
Sliding windows are strategically placed around the histogram peaks and moved vertically across the image. This technique adjusts dynamically to track the curvature of the lane lines based on the accumulated lane line pixels within each window.
A second-order polynomial curve is then fitted to these pixels for both the left and right lane lines. This mathematical modeling is critical as it not only predicts the current lane curvature but also facilitates the estimation of future road curvature.

9) Calculating Curvature and Vehicle Offset:

The polynomial curves are used to compute the radius of curvature for each lane line, providing essential feedback on the road geometry. Additionally, the vehicle's position relative to the lane center is calculated, offering vital navigation data to assist in maintaining lane discipline or for autonomous vehicle steering adjustments.

10) Perspective Transformation and Overlay:

Detected lane lines are projected back onto the original image using an inverse perspective transformation. This step corrects the viewing angles and scales the detected lines to fit the real-world perspective seen by the camera, enhancing the visual integration of the lane lines with the actual road.

11) Display of Results:

Finally, the enhanced frame with lane overlays, along with data on lane curvature and vehicle offset, is displayed. This visualization not only confirms the accuracy of lane detection but also provides crucial driving assistance information.

This advanced lane detection methodology employs a suite of computer vision techniques to create a robust system capable of accurately detecting and visualizing lane boundaries under varying environmental conditions. This comprehensive approach ensures the reliability and effectiveness of the lane detection system in real-time driving scenarios.

IV. Experiments and Results

1. In the project, we explored the performance of three prominent color spaces—**HSV**, **HSL**, and **RGB**—for lane detection. Through implementation and testing, we gained valuable insights into the strengths and limitations of each color space.

HSV Color Space

Implementing HSV color space involved separating the image into its hue, saturation, and value components. Our investigation into the HSV color space revealed its robustness in handling variations in lighting conditions. By isolating the saturation channel, we observed that lane markings could be effectively highlighted, offering reliable detection even in challenging lighting scenarios such as shadows or glare. However, we also encountered limitations, particularly in scenarios with subtle color differentiations in lane markings. In such cases, the hue and saturation components alone sometimes failed to provide sufficient discrimination, leading to potential inaccuracies in detection.

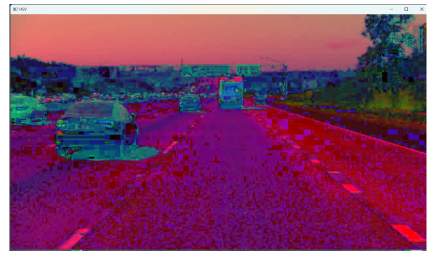

Fig. 4. Output images of HSV color space

RGB Color Space

Similarly, our implementation of the RGB color space allowed us to dissect the image into its red, green, and blue components. Our exploration of the RGB color space highlighted its widespread use and straightforward representation of colors. While RGB provides a fundamental framework for color representation, we noted that it may not inherently offer robustness to changes in lighting conditions. Nevertheless, by integrating additional image processing techniques such as edge detection and morphological operations, we observed improvements in lane detection accuracy. Despite these enhancements, RGB alone may not be as effective in scenarios with significant lighting variations.

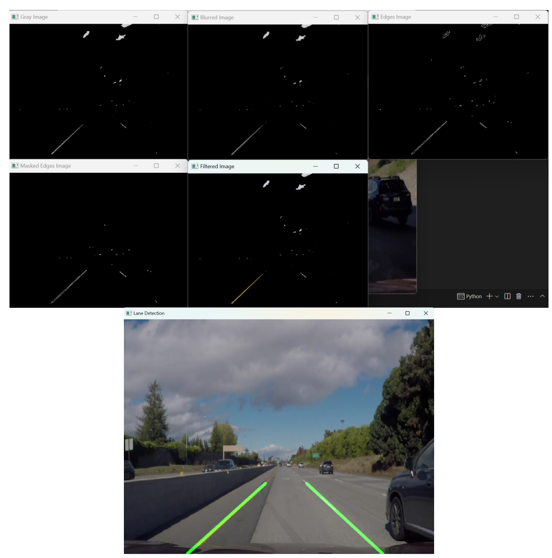

Fig. 5. Output images of RGB color space

HSL Color Space

Among the color spaces examined, HSL emerged as the standout performer in our experiments. By replacing the value component with lightness, HSL provided better control over brightness variations, facilitating clearer discrimination between lane markings and the surrounding road surface. This capability proved invaluable in scenarios with subtle color variations or varying lighting conditions. Additionally, HSL maintained robustness to changes in saturation, ensuring reliable detection across diverse environments.

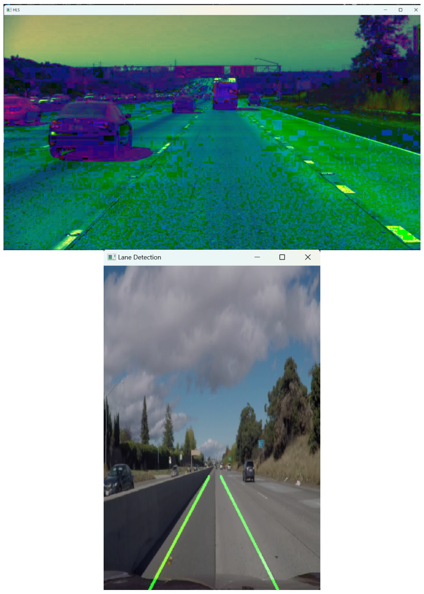

Fig. 6. Output images of HSL color space

While experimenting with multiple color spaces for the most accurate lane detection, we compared the performance of three color spaces: RGB, HSV, and HSL. We discovered a key limitation with RGB. Because it combines color information with intensity, isolating specific lane colors (like white or yellow) proved challenging. This made filtering out non-lane elements based solely on color thresholds difficult. Imagine trying to pick out yellow dandelions in a field on a bright day – the bright sunlight affects how yellow they appear. HSL offered a significant advantage with its separate lightness channel. This separation allowed us to perform color thresholding much more effectively. Think of lightness as a dial controlling brightness. By adjusting this dial, we could isolate lane colors regardless of the overall image brightness. This improved our lane detection in various lighting conditions. For example, we could now confidently identify white lane markings on a scorching day or distinguish them from a dark, wet road surface. This improved accuracy reduced deviations and flickering – the car no longer strayed from the actual lane lines as much. While HSV also provided decent lane detection, surpassing RGB's capabilities, it couldn't quite match the robustness of HSL. In essence, HSL offered the most precise and consistent lane color identification, leading to smoother and more reliable lane detection.

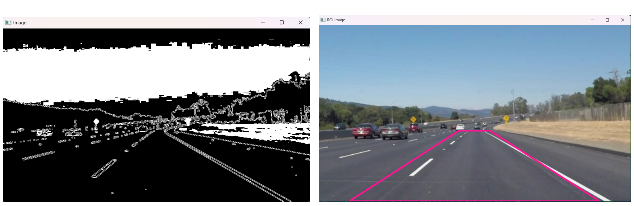

Fig. 7. a) Sobel edge b) Trapezoid view (ROI)

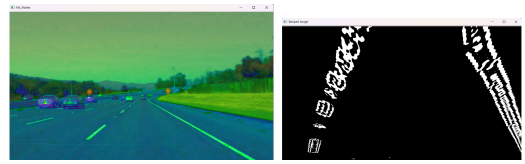

Fig. 8. c) HSL frame d) Wraped frame

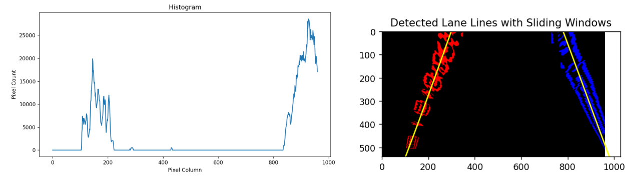

Fig. 9. e) Histogram f) Sliding window frame

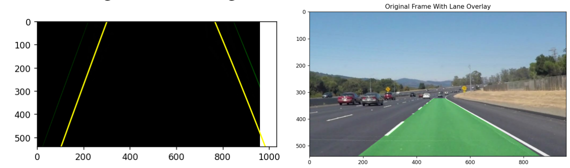

Fig. 10. g) Lane marking h) Overlay lane mapping on frame

Fig. 11. Step wise expiremental output

2. In our advanced lane project, we tackled lane detection using the HSL color space. This space separates colors (Hue), intensity (Saturation), and brightness (Lightness), making it ideal for isolating lane markings from the background compared to 'RGB'.
We begin by converting each video frame to 'HSL'. We then focus on the Lightness channel, applying 'Sobel Edge' Detection(fig7: a) to highlight sharp intensity changes – potential lane edges. Next, independent thresholds are applied to the Saturation and Hue channels, targeting pure white and yellow pixels commonly used for lane markings. This filtering removes distractions like shadows or road color variations. To further refine the image, we perform a 'bitwise AND' operation, essentially combining the filtered channels and eliminating inconsistencies. Camera perspective creates a trapezoidal road view(fig7: b), so we implemented a 'perspective transformation' to create a top-down, rectangular view for better lane detection. With the transformed image, we utilize a histogram(fig9: e) to identify areas with high concentrations of white or yellow pixels, likely corresponding to the lanes. Finally, a sliding window algorithm(fig9: f) scans the image, searching for patterns that match the filtered lane pixels. To further enhance accuracy, we fit a second-degree polynomial curve to the lane pixels. This smooths out irregularities and provides a precise mathematical representation of the lane's trajectory. The result? We can detect and represent lane markings(fig10: g) with green lines overlaid on the original video. More importantly, we calculate lane curvature based on the fitted curves. The left curvature is calculated to be **6.68** meters and the right curvature is **6.4** meters. The center offset is shifted by **-13** cm. This curvature information is crucial for the autonomous vehicle to understand road geometry and adjust steering to stay within lane boundaries. We also calculate the

vehicle's center offset from the lane center, providing real-time data for precise lane positioning. This information is used in a feedback loop to continuously adjust the vehicle's position and maintain safe travel within the lane.

3. In order to enhance accuracy and implement an advanced lane detection system, we explored the implementation of Convolutional Neural Networks (CNNs). These networks have revolutionized lane detection in autonomous driving systems, offering a robust and adaptable solution by effectively identifying lane patterns and structures across diverse driving conditions. Renowned for their prowess in computer vision tasks, CNNs excel in lane detection by leveraging spatial hierarchies in images, enabling them to discern crucial features like edges, shapes, and textures.

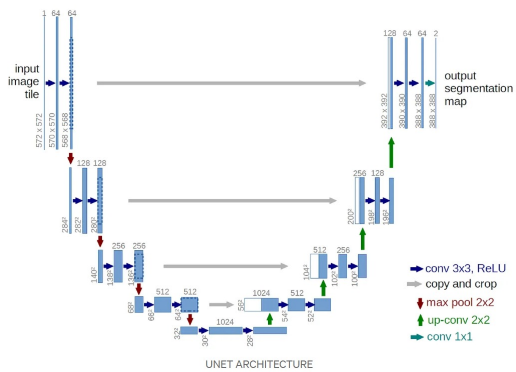

Fig. 12. U-Net architecture

Among CNN architectures, we found U-Net to be particularly effective for segmenting lane lines from high-resolution images. Originally designed for medical image segmentation, U-Net's architecture suits tasks requiring precise localization, such as lane detection. Its contracting and expanding paths adeptly capture context, facilitating accurate localization essential for safety and performance in lane detection.

For our project, we leveraged the TuSimple dataset, which comprises over 6408 road images, partitioned into 3,626 for training, 358 for validation, and 2,782 for testing. This dataset encompasses diverse weather conditions, providing crucial training data for robust model development. Employing the U-Net architecture, we preprocessed the images to meet network input requirements, including resizing and normalization. Despite computational constraints, we trained the network on a trimmed dataset of 100 images, ensuring manageable computation. Our evaluation of the U-Net model's performance on the TuSimple dataset primarily focused on accuracy and lane detection precision. Despite the limitations posed by computational resources, we successfully trained the network and visualized the training images with lane parameters extracted from JSON files. This iterative process lays the groundwork for further enhancements and the eventual real-world deployment of autonomous driving systems with improved lane detection capabilities.

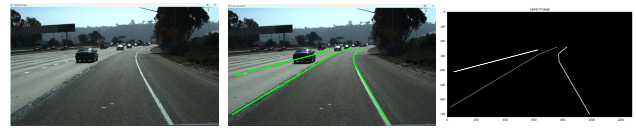

Fig. 13. Visualization of Tu-simple dataset

After training the model, we attempted to assess its performance on an image. Unfortunately, the results fell short of expectations. Upon observing the predicted mask image, it became evident that the model also captured irrelevant features of various objects in the scene, which are not beneficial for accurate lane detection.The program also identified adjacent lanes, possibly due to insufficient training since the model was only trained for 10 epochs and a limited dataset.

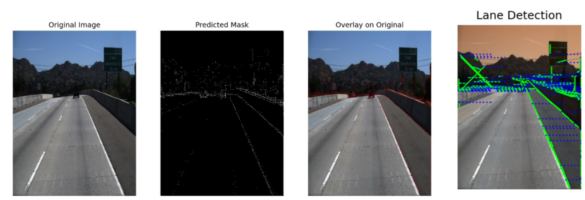

Fig. 14. CNN test image outcome

V. Discussion and Summary

This project encapsulates a synthesis of non-deep learning image processing methods for lane detection. The collective research underscores the potency of traditional algorithms like Gaussian Blurring, Canny Edge Detection, and Hough Transform in diverse lighting conditions and at different times of the day, notably demonstrating optimal performance in full daylight.

Comparative studies further highlight the juxtaposition of non-deep learning approaches with deep learning models, accentuating the trade-offs between computational efficiency and robustness against environmental challenges. Experimental results affirm the reliability of the proposed solution during peak daylight, with noted limitations under low-light conditions and in the presence of shadows.

The project's trajectory aligns with the evolving landscape of autonomous navigation systems, aspiring to bridge the gap between high-performance detection and resource-constrained environments. Future work is poised to address the identified limitations, with a focus on enhancing night-time detection and shadow resilience, ensuring the continuity of safety in all Real-Time driving conditions, Explore the use of advanced CNN architectures, which have demonstrated proficiency in capturing lane features and handling high curvature roads in various conditions.

References

[1] D. Pomerleau, "RALPH: Rapidly adapting lateral position handler," in *Proc. Intell. Veh. Symp.*, 1995, pp. 506–511.

[2] J. H. Kim, S. K. Kim, S. H. Lee, T. M. Lee, and J. Lim, "Lane recognition algorithm using lane shape and color features for vehicle black box," in *International Conference on Electronics, Information and Communication (ICEIC)*, 2018, vol. 2018-Janua, pp. 1–2, doi: 10.23919/ELINFOCOM.2018.8330549.

[3] S. I. A, K. R, H. Shanmugasundaram, B. p. A, K. R and M. B. J, "Lane Detection Using Deep Learning Approach," in *2022 1st International Conference on Computational Science and Technology (ICCST)*, Chennai, India, 2022, pp. 945-949, doi: 10.1109/ICCST55948.2022.10040402.

[4] M. Jefri Muril, N. H. Abdul Aziz, H. A. Ghani and N. A. Ab Aziz, "A Review on Deep Learning and Nondeep Learning Approach for Lane Detection System," in *2020 IEEE 8th Conference on Systems, Process and Control (ICSPC)*, Melaka, Malaysia, 2020, pp. 162-166, doi: 10.1109/ICSPC50992.2020.9305788.

[5] M. Doshi, H. Shah and N. Katre, "ROI based real time straight lane line detection using Canny Edge Detector and masked bitwise operator," in *2022 IEEE Bombay Section Signature Conference (IBSSC)*, Mumbai, India, 2022, pp. 1-6, doi: 10.1109/IBSSC56953.2022.10037363.

[6] Y. Kortli, M. Marzougui, B. Bouallegue, J. S. C. Bose, P. Rodrigues and M. Atri, "A novel illumination-invariant lane detection system," in *2017 2nd International Conference on Anti-Cyber Crimes (ICACC)*, Abha, Saudi Arabia, 2017, pp. 166-171, doi: 10.1109/Anti-Cybercrime.2017.7905284.

[7] N. bin Abdul Razak, M. Z. bin Mazlan, J. B. Johari, S. A. Bin Che Abdullah and N. K. Mun, "A Lane Detection Using Image Processing Technique for Two-Lane Road," in *2022 IEEE 10th Conference on Systems, Process Control (ICSPC)*, Malacca, Malaysia, 2022, pp. 214-219, doi: 10.1109/ICSPC55597.2022.10001801.